



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/559,568	12/05/2005	Jurg Portmann	126067	2964
25944	7590	10/17/2007		
OLIFF & BERRIDGE, PLC			EXAMINER	
P.O. BOX 320850			HUR, ECE	
ALEXANDRIA, VA 22320-4850			ART UNIT	PAPER NUMBER
			4135	
			MAIL DATE	DELIVERY MODE
			10/17/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/559,568

Applicant(s)

PORTMANN, JURG

Examiner

ECE HUR

Art Unit

4135

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 05 December 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 05 December 2005 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☒ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 03/06/2006.
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_.

### **DETAILED ACTION**

This action is responsive to application filed on November 5, 2005 and IDS filed on March 6, 2006, in which Claims 1-20 are presented for examination. This application is a new PCT National Stage application of PCT/EP04/50776 that was filed on May 12, 2004. Applicant is claiming foreign priority for the application EPO 03012822.7 filed on June 5, 2003.

#### ***Status of Claims***

Claims 1-20 are pending in the case. Claims 1 and 17 are the independent Claims.

Claims 1-16 are rejected under 35 U.S.C. 112, second paragraph.

Claims 14-16 are rejected under 35 U.S.C. 101.

Claims 1-20 under 35 U.S.C. 102(e).

#### ***Information Disclosure Statement Acknowledgement***

The information disclosure statement filed on March 6, 2006 is in compliance with the provisions of 37 CFR 1.97, 1.98 and MPEP § 609. It has been placed in the application file, the information referred to therein has been considered as to the merits.

#### ***Priority Acknowledgement***

Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d). Receipt is acknowledged of certified copy of (EPO) 03012822.7 filed on June 5, 2003 submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

***Specification Objection***

The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-16 are rejected under 35 U.S.C. 112, second paragraph, as being failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention, because the Claims recite "and/or".

***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 14-16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter, specifically directed towards Software per se.

Regarding Claims 14-16, Claims 14-16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter and claiming "Software" per se. Software is functional descriptive material that can be considered statutory only if it is both functional and clearly embodied on a computer readable medium and designed to support specific data manipulation function. When functional descriptive material is recorded on a computer-readable medium it will become structurally and functionally interrelated the medium and will be statutory in most cases since the use of technology permits the function of the descriptive material to be realized. See *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031, 1035 (Fed. Cir 1994) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQd at 1759. A Software structure is functional if the specific arrangement of data enables a computer to accomplish useful result arising from the arrangement of the data in the software. However, only computer readable medium executed instruction by a processor could be statutory, it is not clearly defined as being embodied in a computer readable medium as executed instruction and is therefore not statutory. See *Warmerdam*, 33 F.3d at 1360, 31 USPQ2d at 1759.

***Claim Rejections - 35 USC § 102***

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-20 are rejected under 35 U.S.C. 102(e) as being anticipated by  
Prinzing, US 6,496,202.

Regarding Claim 1, Prinzing in US 6,496,202 discloses the claimed aspect of method for generating a user interface of a network node, whereas an application is structured into a core application part responsible for handling data objects and a viewer/controller application part responsible for displaying said data and initiating actions on said data and said viewer/controller application part is formed by said user interface, wherein a graphical user interface (GUI) is customized that represents the view in a Model/View/Controller architecture. The model contains an application's data and one or more structural components that are used to identify the GUI components of the view. A factory builds the GUI using characteristics of the structural components to identify a set of GUI components to create the view. (Prinzing, Column 4, lines 58-65).

Prinzing discloses the claimed aspect of characterized in that a screen mask creating module (240) for creating dynamically a screen mask of said user interface retrieves screen mask configuration data (320) and widget configuration

Art Unit: 4135

data (310) over a network which are stored on a central processing unit, in that a widget creating module (230) generates at least one user interface component (10-18; 410) on the basis of one or more component patterns (411, 412) of the widget configuration data (310) and stores the at least one user interface component (10-18; 410) by means of a widget cache (210), in that a screen mask of said user interface is generated by said screen mask creating module (240) in FIGURE 9 and FIGURE 2, wherein the factory can build a view using GUI components from multiple look-and-feel standards (e.g., Motif, Presentation Manager, Microsoft Windows, or Macintosh). A GUI component can be added to the set of GUI components known to the factory. The factory is instructed to use a GUI component for a given characteristic of a structural component. (Prinzing, Column 5, lines 1-5).

Prinzing discloses the claimed aspect of said screen mask comprises at least one component which is a component out of said components comprising the widget configuration data, and in that said at least one component of said created screen mask is assigned to at least one data object and/or dynamic of said components assigned to said screen mask based upon a user action on a user interface component and/or a data object in FIGURE 6, wherein model 606 contains the data associated with a software application. View 604(Application Interface) constitutes the portion of the GUI that is visible to a user. View 604 includes a plurality of view fragments (e.g., view fragment 654-662) created by Factory 602. Information contained in Model 606 is used to identify some or all of the view fragments that comprise View 604. Factory 602 is asked to create a

Art Unit: 4135

view fragment from a set of view components known to Factory 602. Factory 602 can receive a request to create a view fragment based on the contents of Model 606. A view fragment that is created by Factory 602 may wish to delegate (e.g., some functionality or some of its responsibilities) to one or more child view fragments. In that case, the view fragment requests that Factory 602 create the child view fragment(s). (Prinzing, Column 7, lines 52-65). Furthermore, the GUIFactory 102, has GUI components such as scroll bar, window and button. (Prinzing, Column 2, lines 7-9).

Regarding Claims 2 and 3, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of characterized in that said screen mask of said user interface is dynamically generated by said screen mask creating module (240) and screen mask configuration data and/or widget configuration data are at least partially changed dynamically based upon one or more user actions, wherein an abstract factory can be used, for example, to allow an application developer to write a software application that contains references to the GUI components of a generic GUI library. At run time, the GUI library is specified and the GUI component references are resolved to refer to GUI components in the specified GUI library. Using this technique, a software application can be written to use one GUI library's GUI components (e.g., Motif GUI components) in one runtime environment and another GUI library's GUI components (e.g., Presentation Manager GUI components) in another runtime environment. (Prinzing, Column 1,



lines 47-57). Furthermore, one of the concrete factories is instantiated at runtime based on the look-and-feel standard that is specified (e.g., as a runtime parameter). If, for example, the look-and-feel standard is determined at runtime to be Motif, an instance of MotifFactory is instantiated. The second code statement provided above would have the effect of invoking the CreateScrollBar method or operation of the MotifFactory object which would create a scroll bar having the Motif look-and-feel. (Prinzing, Column 2, lines 20-27).

Regarding Claims 4, most of the limitations have been met in the rejection of Claims 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect characterized in that said screen mask configuration data (320) is retrieved (S220), said screen mask configuration data (320) is parsed (S220) to obtain type information about said at least one component (10-18; 410) and to obtain individual settings of said at least one component (10-18; 410), said at least one component (10-18; 410) is created (S230) by obtaining (S231) said at least one component (10-18; 410) on the basis of at least one component pattern (411, 412) corresponding to said type information and said individual settings is applied (S232) onto said at least one component (10-18; 410) and including said at least one component (10-18; 410) into said screen mask in FIGURE 6, wherein model 606 contains the data associated with a software application. View 604 constitutes the portion of the GUI that is visible to a user. View 604 includes a plurality of view fragments (e.g., view fragment 654-662) created by Factory 602. Information contained in Model 606 is used to identify some or all of the view

Art Unit: 4135

fragments that comprise View 604. Factory 602 is asked to create a view fragment from a set of view components known to Factory 602. Factory 602 can receive a request to create a view fragment based on the contents of Model 606. (Prinzing, Column 7, lines 52-61). Furthermore, parsing logic and application interface are illustrated.

Regarding Claim 5, most of the limitations have been met in the rejection of Claim 4. See details for Claim 4 rejection. Prinzing discloses the claimed aspect least one component (10-18; 410) are requested from a component pattern repository (210), which caches at least one component pattern (411, 412), said at least one component pattern (411, 412) are identified corresponding to said type information, and said at least one component (10-18; 410) are derived from said at least one identified component pattern (411, 412) in FIGURE 6, wherein Model-Driven Factory 602 has different object classes more specifically Factory 602 creates View 604 by creating instances of object classes that are known to it. (Prinzing, Column 12, lines 58-61).

Furthermore, an object class can be added to or removed from the set of object classes known to Factory 602. FIG. 7 provides an example of a create object instance process flow according to an embodiment of the invention. In step 702, a create request is sent to Factory 602 along with the Element of Model 606 to which the new object instance is associated. At step 704, the Attributes object instance associated with the Element instance specified in the create

request can be queried to obtain the attributes associated with Element. Factory 602 uses the attributes associated with the Element to determine what object class to instantiate at step 706. For example, Factory 602 can use the "value" associated with the NAME attribute to identify an object class (e.g., that has the same name as the "value"). (Prinzing, Column 12-13, lines 61-68, 1-6).

Regarding Claim 6, most of the limitations have been met in the rejection of Claim 5. See details for Claim 5 rejection. Prinzing discloses the claimed aspect of component pattern repository (210) is initialized by retrieving (S110) said widget configuration data (310), which comprises widget configuration data about at least one component pattern (411, 412), parsing (S120) said widget configuration data (310), creating (S130) said at least one component pattern, and storing said at least one component pattern in said component pattern repository (210) in FIGURE 6, wherein a mapping can be provided to Factory 602 when a new class is added to its set of object classes that specifies the object class that is to be instantiated for a particular "value" of the NAME attribute. In both cases, the attributes of an Element specify characteristics of Model 606 that are used to identify a view object class instance (i.e., a view fragment) to be created by Factory 602. (Prinzing, Column 13, lines 6-12). Furthermore, parsed pattern is illustrated in Model-Driven Factory 602.

Regarding Claim 7, most of the limitations have been met in the rejection of Claim 5. See details for Claim 5 rejection. Prinzing discloses the claimed aspect

Art Unit: 4135

of component pattern repository (210) contains statically said at least one component pattern (41.1, 412) during runtime of said application in FIGURE 6, wherein the abstract factory uses factory and product abstract classes to define the methods that are implemented by their respective concrete classes. An application does not directly instantiate classes. Instead, objects can be instantiated indirectly as illustrated by the second code statement. The application effectively uses a pointer to the object to be instantiated that is resolved at runtime. (Prinzing, Column 2, lines 40-46).

Regarding Claim 8, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of least one component (10-18; 410) are obtained by requesting said at least one component (10-18; 410), retrieving (S110) a widget configuration data (310), which comprises widget configuration data about at least one component pattern (411, 412), identifying said widget configuration data about said at least one component pattern (411, 412) corresponding to said extracted type information, parsing (S120) said identified widget configuration information, creating (S130) said at least one component pattern, deriving said at least one component (10-18; 410) from said at least one component pattern (411, 412) in FIGURE 6, wherein Model-Driven Factory 602 has different object classes more specifically Factory 602 creates View 604 by creating instances of object classes that are known to it. (Prinzing, Column 12, lines 58-61).

Furthermore, an object class can be added to or removed from the set of object classes known to Factory 602. FIG. 7 provides an example of a create object instance process flow according to an embodiment of the invention. In step 702, a create request is sent to Factory 602 along with the Element of Model 606 to which the new object instance is associated. At step 704, the Attributes object instance associated with the Element instance specified in the create request can be queried to obtain the attributes associated with Element. Factory 602 uses the attributes associated with the Element to determine what object class to instantiate at step 706. For example, Factory 602 can use the "value" associated with the NAME attribute to identify an object class (e.g., that has the same name as the "value"). (Prinzing, Column 12-13, lines 61-68, 1-6).

Regarding Claim 9, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of at least one component (10-18; 410) are obtained by cloning said at least one component pattern (411, 412) to obtain said at least one component (10-18; 410) in FIGURE 6, wherein on Application Interface 604 chosen pattern are illustrated. Furthermore, Prinzing discloses the claimed aspect of cloning, wherein copyAttributes(Attributes) Copies the locally defined attribute set to another Attributes instance. (Prinzing, Column 12, lines 16-19).

Regarding Claim 10, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of widget configuration data (310) comprises default widget configuration information about said at least one component pattern (411, 412) such that components (10-18; 410) obtained from said least one component pattern (411, 412) have default settings valid for substantially all components used in the user interface in FIGURE 6, wherein element 670 can be determined using a default root element instance, or by specifying a root element instance. The getDefaultRootElement method of Document 666 can be used to return a default root element. For example, the getDefaultRootElement can return an instance of an HTML element object class that is the default root element for the tree structure created for an HTML document. (Prinzing, Column 11, lines 25-34).

Regarding Claim 11, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of screen mask configuration data (320) comprises screen mask configuration data about at least one component (10-18; 410) to adapt said component (10-18; 410), which is obtained from said corresponding component pattern (411, 412), to requirements presupposed by said screen mask to be created, wherein attributes are fetched from the collection of attributes this element contains and element is fetched at the given index. (Prinzing, Column 11, lines 40-42).

Regarding Claims 12 and 13, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect screen mask configuration data (320) is an XML-encoded screen mask configuration data, which is based on a screen mask document type description (DTD; 340) and said widget configuration (310) is an XML-encoded widget configuration, which is based on a widget document type description (DTD; 330), wherein the structure in FIGURE 6 can be expressed using a using a language such as Standard Generalized Markup Language (SGML). SGML is a meta-language from which other languages can be created. SGML uses "tags" associated with information contained in a document to identify an element and, possibly, attributes associated with the element. (Prinzing, Column 9, lines 14-18). Furthermore, the Hypertext Markup Language (HTML) used to define Web pages is an example of a language created from SGML. The following is an example of an HTML document that contains tags for defining structure: <HTML> . . . <INPUT TYPE=text NAME=inputfield1 VALUE=""> . . . (Prinzing, Column 9, lines 20-25). Applicant should duly note that both SGML and XML are "meta" languages because they are used for defining markup languages.

Regarding Claims 14, 15 and 16, most of the limitations have been met in the rejection of Claim 1. See details for Claim 1 rejection. Prinzing discloses the claimed aspect of software tool, computer program product for establishing a user interface (GUI), comprising program portions, loadable program portions that are sections stored on a computer readable medium for carrying out the

operations of Claim 1, when said program is implemented in a computer program for being executed on a microprocessor-based device, processing device, a terminal device or a network device in FIGURE 9, wherein an embodiment of the invention can be implemented as computer software in the form of computer readable program code executed on a general purpose computer such as computer 900 illustrated. Furthermore, Network link 921 provides data communication. (Prinzing, Column 17, lines 3-5).

Regarding Claim 17, Prinzing achieves the claimed aspect of a terminal device, wherein an apparatus for generating a graphical user interface is disclosed. (Prinzing, Column 6, lines 30-32). Prinzing discloses the claimed aspect of an user interface, which is operable by a user to operate an application executed by said terminal device, which comprises a screen mask creating component (240) for creating dynamically a screen mask of said user interface (GUI), wherein a graphical user interface (GUI) is customized that represents the view in a Model/View/Controller architecture. The model contains an application's data and one or more structural components that are used to identify the GUI components of the view. A factory builds the GUI using characteristics of the structural components to identify a set of GUI components to create the view. (Prinzing, Column 4, lines 58-65).

Prinzing discloses the claimed aspect of a retrieval component (260, 270) for retrieving a screen mask configuration data (320) and widget configuration data (310), which comprises configuration data about at least one component (10-18;



Art Unit: 4135

410), in FIGURE 9 and FIGURE 2, wherein the factory can build a view using GUI components from multiple look-and-feel standards (e.g., Motif, Presentation Manager, Microsoft Windows, or Macintosh). A GUI component can be added to the set of GUI components known to the factory. The factory is instructed to use a GUI component for a given characteristic of a structural component. (Prinzing, Column 5, lines 1-5).

Prinzing discloses the claimed aspect of a parsing component (250, 230, 240) for parsing said screen mask configuration data (320) to obtain type information about said at least one component (10-18; 410) and to obtain individual settings of said at least one component (10-18; 410), and for parsing said widget configuration data (310) to obtain one or more component patterns (411, 412), wherein in FIGURE 6, wherein model 606 contains the data associated with a software application. View 604 constitutes the portion of the GUI that is visible to a user. View 604 includes a plurality of view fragments (e.g., view fragment 654-662) created by Factory 602. Information contained in Model 606 is used to identify some or all of the view fragments that comprise View 604. Factory 602 is asked to create a view fragment from a set of view components known to Factory 602. Factory 602 can receive a request to create a view fragment based on the contents of Model 606. (Prinzing, Column 7, lines 52-61). Furthermore, parsing logic and application interface are illustrated.

Prinzing discloses the claimed aspect of a widget creating component (230) for obtaining said at least one component (10-18; 410) on the basis of at least one component pattern (411, 412) corresponding to said type information and for

Art Unit: 4135

applying said individual settings onto said at least one component (10-18; 410), and a linking component (430) for linking said at least one component (10-18; 410) to at least one data object (460, 465) in FIGURE 2, wherein a GUI creator creates Concrete Creator by type MotifUI and Concrete Creator B by type PMUI. Furthermore, in FIGURE 6 an application interface is illustrated with linked objects.

Regarding Claim 18, most of the limitations have been met in the rejection of Claim 17. See details for Claim 17 rejection. The rejection for Claim 5 applies to Claim 18. See the rejection details for Claim 5.

Regarding Claim 19, most of the limitations have been met in the rejection of Claim 17. See details for Claim 17 rejection. Prinzing achieves the claimed aspect of a terminal device, wherein an apparatus for generating a graphical user interface is disclosed. (Prinzing, Column 6, lines 30-32). The rejection for Claim 6 applies to Claim 19. See the rejection details for Claim 6.

Regarding Claim 20, most of the limitations have been met in the rejection of Claim 17. See details for Claim 17 rejection. Prinzing discloses the claimed aspect of retrieval component (260, 270) for retrieving a component configuration (310), which comprises component configuration information about at least one component pattern (411, 412), an identification component (240) for identifying said component configuration information about said at least one component

Art Unit: 4135

pattern (411, 412) corresponding to said extracted type information, and a parsing component (250, 230) for parsing said identified component configuration information, wherein said widget creating component (230) is adapted to create said at least one component pattern (411, 412) and to derive said at least one component (10-18, 510) from said at least one component pattern (411, 412) in FIGURE 6, wherein Model-Driven Factory 602 has different object classes more specifically Factory 602 creates View 604 by creating instances of object classes that are known to it. (Prinzing, Column 12, lines 58-61).

Furthermore, an object class can be added to or removed from the set of object classes known to Factory 602. FIG. 7 provides an example of a create object instance process flow according to an embodiment of the invention. In step 702, a create request is sent to Factory 602 along with the Element of Model 606 to which the new object instance is associated. At step 704, the Attributes object instance associated with the Element instance specified in the create request can be queried to obtain the attributes associated with Element. Factory 602 uses the attributes associated with the Element to determine what object class to instantiate at step 706. For example, Factory 602 can use the "value" associated with the NAME attribute to identify an object class (e.g., that has the same name as the "value"). (Prinzing, Column 12-13, lines 61-68, 1-6).

### ***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- 1) Stucka, et al., US 5596702, 01/21/1997, "Method and system for dynamically sharing user interface displays among a plurality of application program".
- 2) Purple, et al., US 5745115, 04/28/1998, "Graphical user interface having a shared menu bar for opened applications".
- 3) Beer, US 5,793,368, 08/11/1998, "Method for dynamically switching between visual styles".
- 4) Sanderson, Richard A., US 20020101448 A1, 08/01/2002, "Generating a declarative user interface".
- 5) Davis, Peter E. et al., US 20020133516 A1, 09/19/2003, "Method and apparatus for end-to-end content publishing system using XML with an object dependency graph".
- 6) Dean, Sara Elo, et al., US 20020152244 A1, 10/17/2002, "Method and apparatus to dynamically create a customized user interface based on a document type definition".
- 7) Davidov, Eran et al., US 20030237050 A1, 12/25/2003, "Markup compiler that outputs MIDlets".

8) Davis, et al., US 7,076,728, 07/11/2006, "Method and apparatus for end-to-end content publishing system using XML with an object dependency graph".

9) Whittenberger, et al., US 7,269,795, 10/11/2007, "Dynamically generated user interface for business application integration".

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ECE HUR whose telephone number is 571 270-1972. The examiner can normally be reached on MONDAY-THURSDAY 7:30 AM - 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, FRANTZ COBY can be reached on (571) 272-4017. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service

Art Unit: 4135

Representative or access to the automated information system, call 800-786-

9199 (IN USA OR CANADA) or 571-272-1000.

Ece Hur  
E.H./e.h.

October 14, 2007

  
**FRANK COBY**  
**SUPERVISORY PATENT EXAMINER**